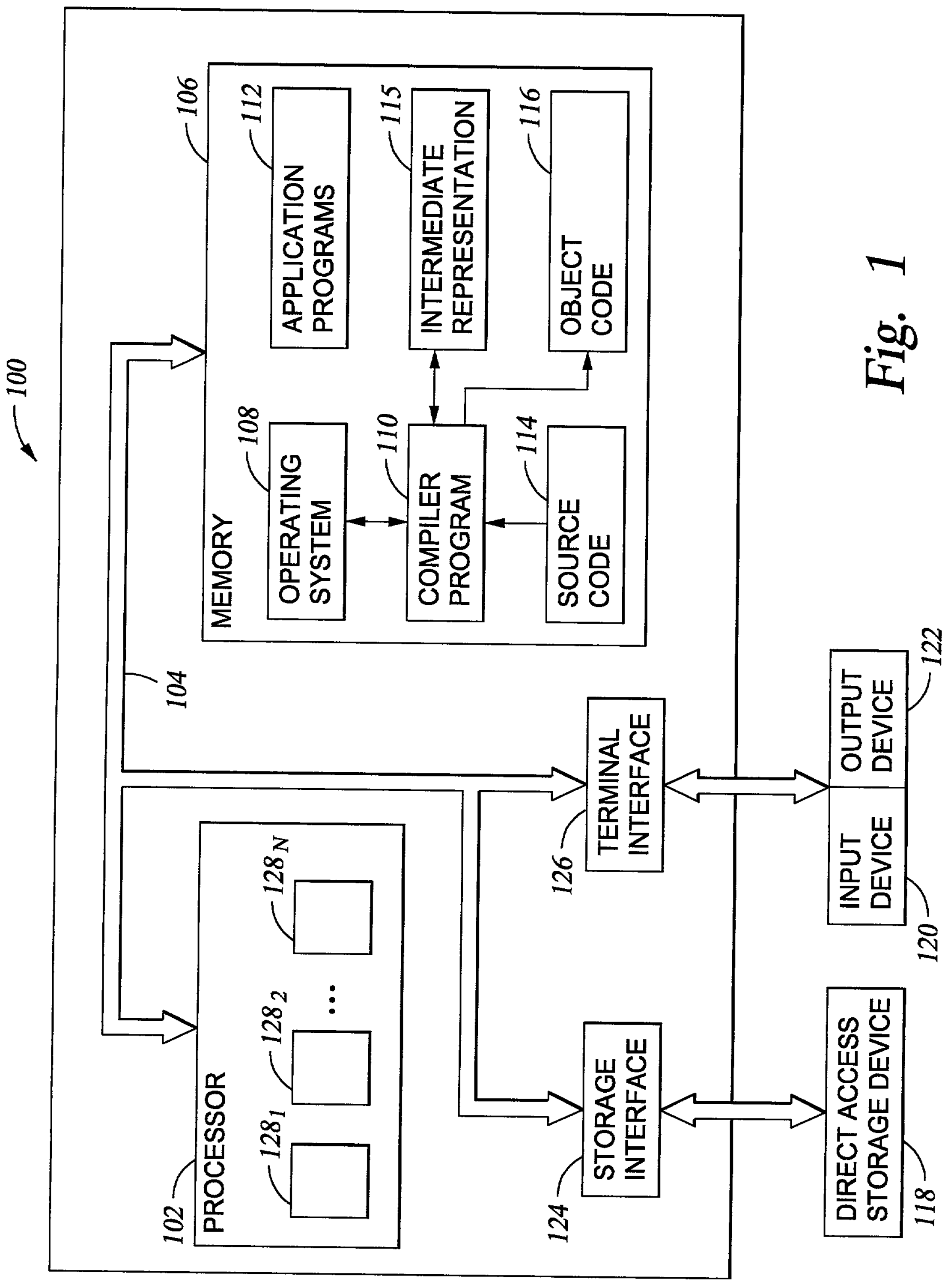

100
MEMORY
106
APPLICATION PROGRAMS
112
INTERMEDIATE REPRESENTATION
115
OBJECT CODE
116
OPERATING SYSTEM
108
COMPILER PROGRAM
110
SOURCE CODE
114
104
PROCESSOR
102
128₁
128₂
128ₙ
TERMINAL INTERFACE
126
STORAGE INTERFACE
124
INPUT DEVICE
120
OUTPUT DEVICE
122
DIRECT ACCESS STORAGE DEVICE
118

Fig. 1

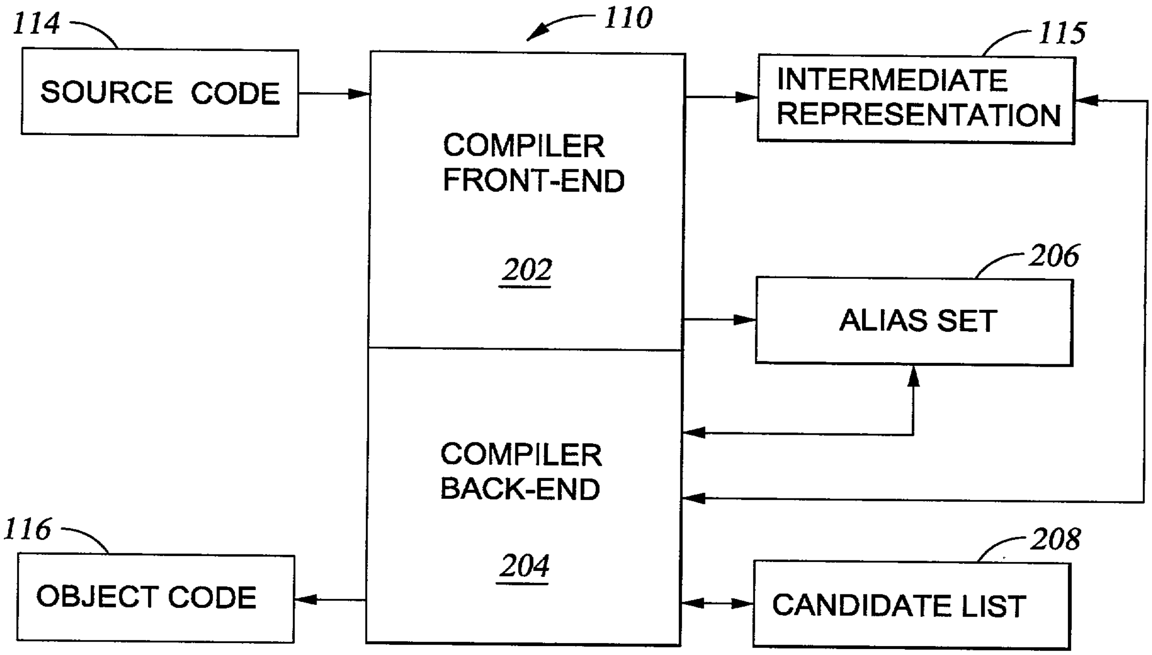
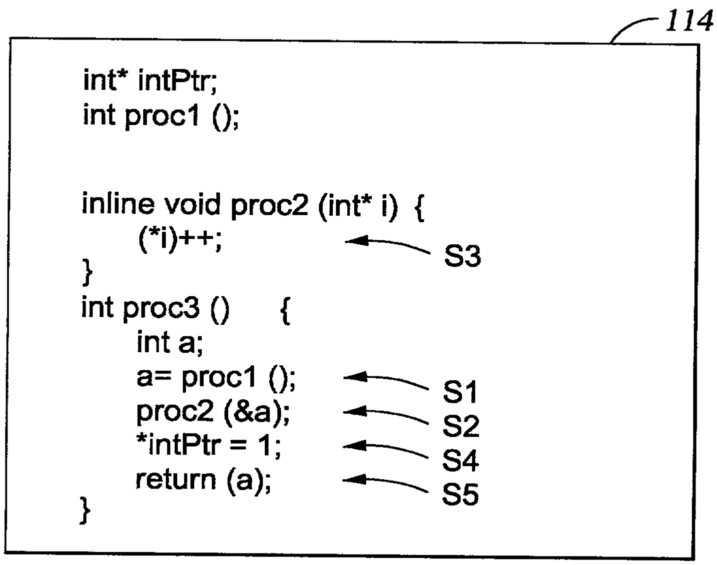


Fig. 2

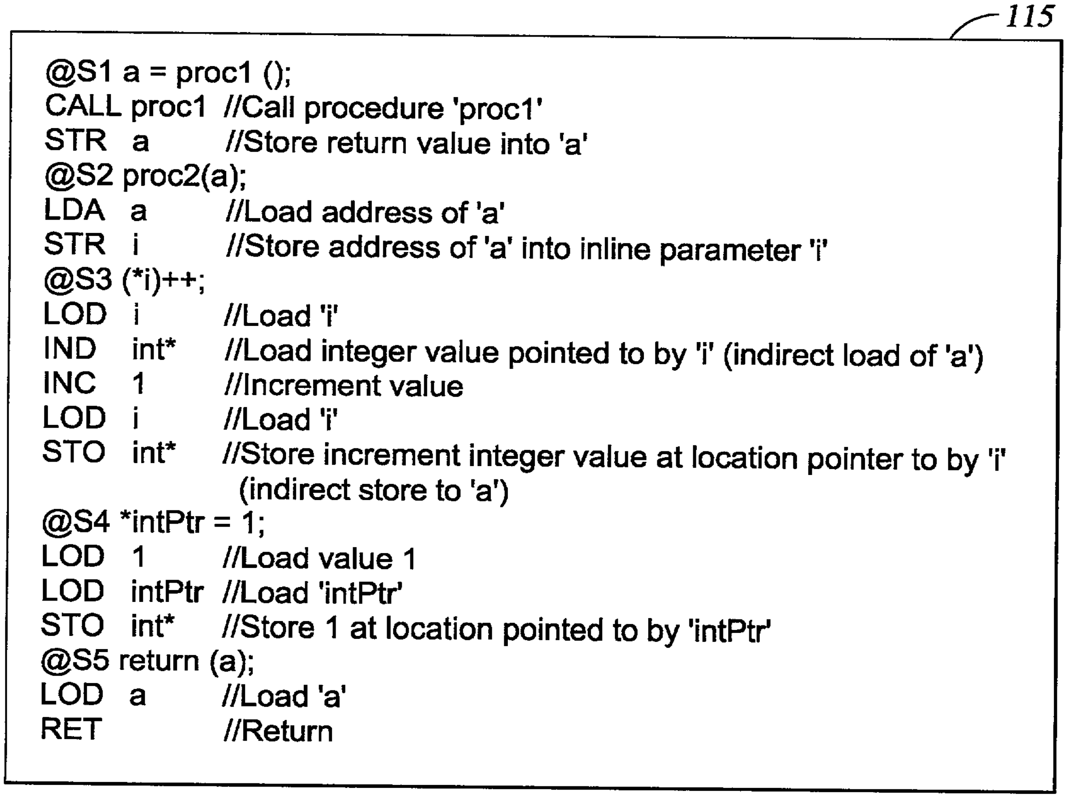


```
int* intPtr;
int proc1 ();


inline void proc2 (int* i) {
     (*i)++;
}
int proc3 ()    {
     int a;
     a= proc1 ();
     proc2 (&a);
     *intPtr = 1;
     return (a);
}
```

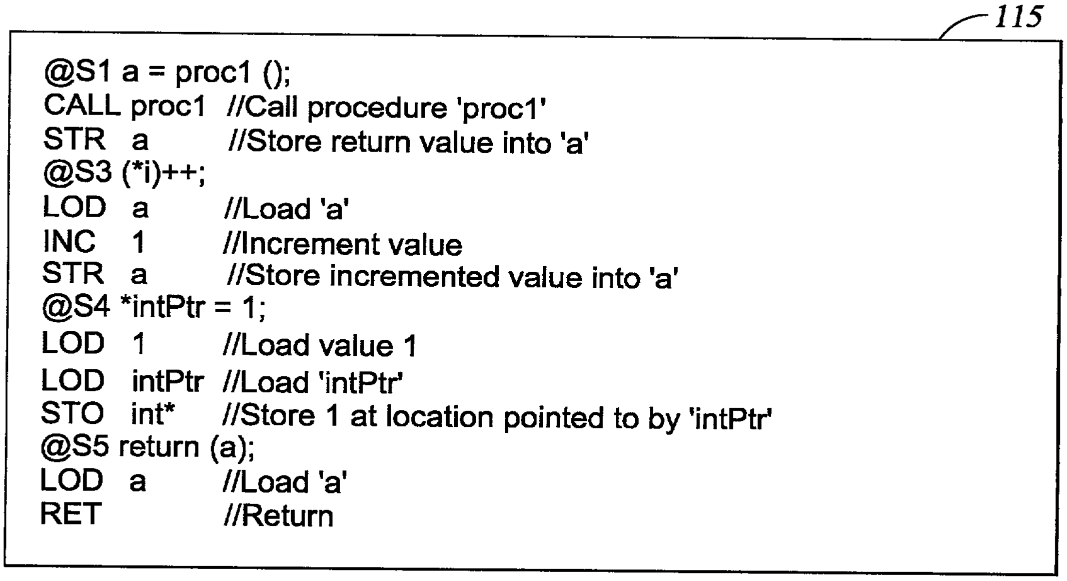
114
S3
S1
S2
S4
S5

Fig. 3



```
@S1 a = proc1 ();
CALL proc1  //Call procedure 'proc1'
STR  a      //Store return value into 'a'
@S2 proc2(a);
LDA  a      //Load address of 'a'
STR  i      //Store address of 'a' into inline parameter 'i'
@S3 (*i)++;
LOD  i      //Load 'i'
IND  int*   //Load integer value pointed to by 'i' (indirect load of 'a')
INC  1      //Increment value
LOD  i      //Load 'i'
STO  int*   //Store increment integer value at location pointer to by 'i'
             (indirect store to 'a')
@S4 *intPtr = 1;
LOD  1      //Load value 1
LOD  intPtr //Load 'intPtr'
STO  int*   //Store 1 at location pointed to by 'intPtr'
@S5 return (a);
LOD  a      //Load 'a'
RET         //Return
```

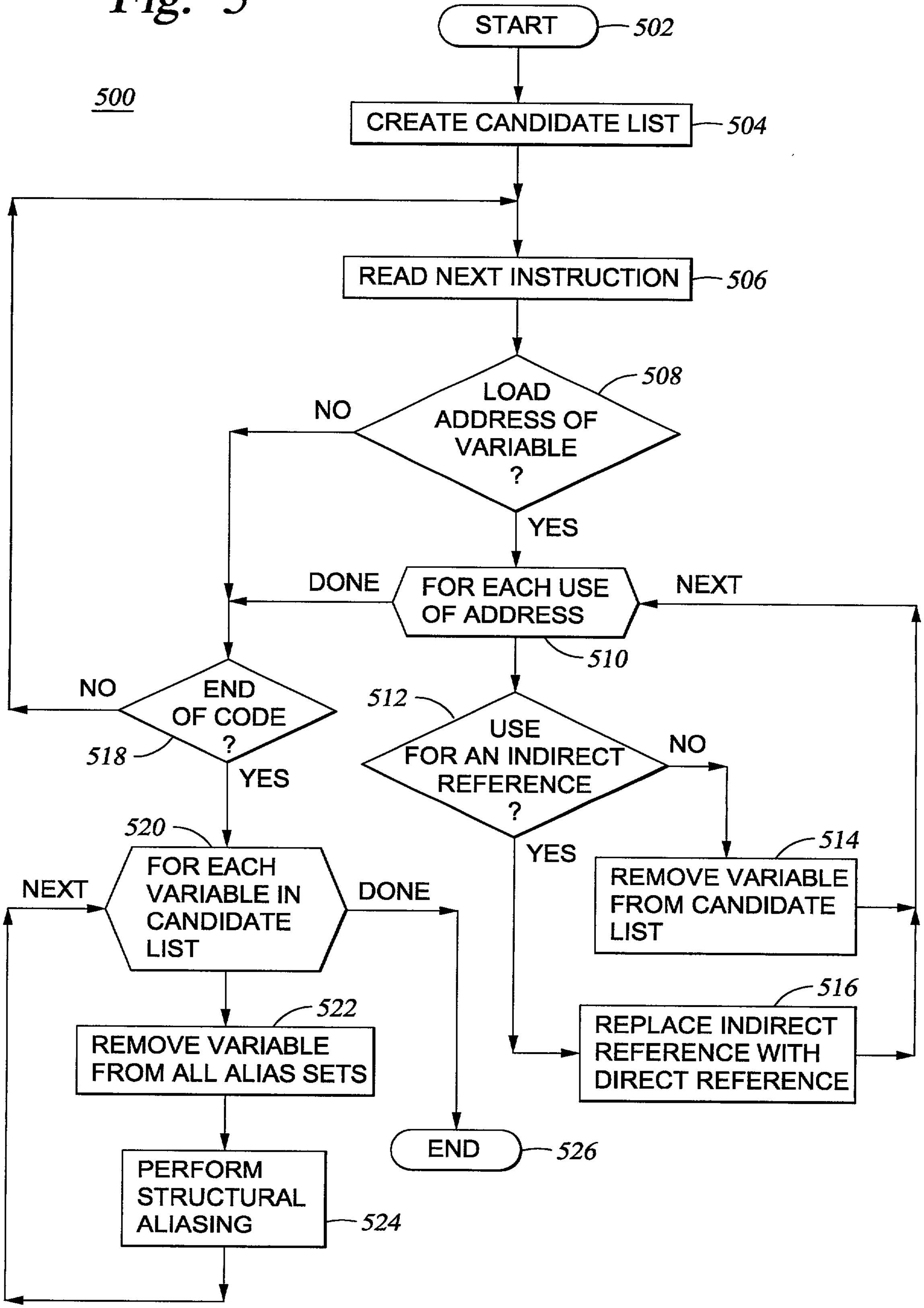
Fig. 4



```
@S1 a = proc1 ();
CALL proc1  //Call procedure 'proc1'
STR  a      //Store return value into 'a'
@S3 (*i)++;
LOD  a      //Load 'a'
INC  1      //Increment value
STR  a      //Store incremented value into 'a'
@S4 *intPtr = 1;
LOD  1      //Load value 1
LOD  intPtr //Load 'intPtr'
STO  int*   //Store 1 at location pointed to by 'intPtr'
@S5 return (a);
LOD  a      //Load 'a'
RET         //Return
```

Fig. 6

Fig. 5


500
START
502
CREATE CANDIDATE LIST
504
READ NEXT INSTRUCTION
506
LOAD ADDRESS OF VARIABLE ?
508
NO
YES
FOR EACH USE OF ADDRESS
510
DONE
NEXT
END OF CODE ?
518
NO
YES
USE FOR AN INDIRECT REFERENCE ?
512
NO
YES
REMOVE VARIABLE FROM CANDIDATE LIST
514
REPLACE INDIRECT REFERENCE WITH DIRECT REFERENCE
516
FOR EACH VARIABLE IN CANDIDATE LIST
520
NEXT
DONE
REMOVE VARIABLE FROM ALL ALIAS SETS
522
PERFORM STRUCTURAL ALIASING
524
END
526